

BIRCH, STEWART, KOLASCH & BIRCH, LLP

INTELLECTUAL PROPERTY LAW

8110 GATEHOUSE ROAD
SUITE 500 EAST
FALLS CHURCH, VA 22042
USA

(703) 205-8000

FAX: (703) 205-8050
(703) 698-8590 (G IV)

e-mail: mailroom@bskb.com
web: http://www.bskb.com

06/22/99
TERRELL C. BIRCH
RAYMOND C. STEWART
JOSEPH A. KOLASCH
JAMES M. SLATTERY
BERNARD L. SWEENEY*
MICHAEL K. MUTTER
CHARLES GORENSTEIN
RALD M. MURPHY, JR.
EDWARD R. SVENSSON
ERRY L. CLARK
ANDREW D. MEIKLE
ARC S. WEINER
J. McKINNEY MUNCY
ROBERT J. KENNEY
JOSEPH FARACI
ONALD J. DALEY
DEN W. BAILEY
ON A. CASTELLANO, III
SENIOR COUNSEL:
ANTHONY L. BIRCH
OF COUNSEL:
HERBERT M. BIRCH
(1905-1996)
ELLIOT A. GOLDBERG*
WILLIAM L. GATES*
EDWARD H. VALANCE
RUPERT J. BRADY (RET)*

*ADMITTED TO A BAR OTHER THAN VA.

06/22/99
GARY D. YACUBO
THOMAS S. ADAMS
MICHAEL R. DAMMAR
JAMES T. ELLER, JR.
SCOTT L. LOWE
JOSEPH H. KIM, PH.D.
RICHARD S. MYERS, JR.*
MARY ANN CAPRIA
MARK J. NUEL, PH.D.
ROBERT V. RACUNAS
DARIN E. BARTHOLOMEW*
D. RICHARD ANDERSON
PAUL C. LEWIS
JERRY W. HOGGE
REG. PATENT AGENTS:
FREDERICK R. HANDREN
ANDREW J. TELESZ, JR.
MARYANNE LIOTTA, PH.D.
MAKI HATSUMI
MIKE S. RYU
W. KARL RENNER
CRAIG A. McROBBIE
GARTH M. DAHLEN, PH.D.
LAURA C. LUTZ
ROBERT E. GOOZNER, PH.D.
HYUNG N. SOHN

Date: June 22, 1999

Docket No.: 2565-0175P

Assistant Commissioner for Patents
Box PATENT APPLICATION
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): YAMAGUCHI, Tomohisa

For: A SYSTEM OF DYNAMIC MODULE CONFIGURATION AND A METHOD
THEREOF

Enclosed are:

X A specification consisting of 21 pages

X 12 sheet(s) of Formal drawings

X An assignment of the invention

X Certified copy of Priority Document(s)

X Executed Declaration X Original Photocopy

 A verified statement to establish small entity status under 37
CFR 1.9 and 37 CFR 1.27

 Preliminary Amendment

X Information Disclosure Statement, PTO-1449 and reference(s)

Other _____

The filing fee has been calculated as shown below:

LARGE ENTITY				SMALL ENTITY	
FOR	NO. FILED	NO. EXTRA	RATE FEE		RATE FEE
BASIC FEE	***** ***** *****	***** ***** *****	***** ***** \$760.00 *****	or	***** ***** \$380.00 *****
TOTAL CLAIMS	16 - 20 =	0	x18 =\$ 0.00	or	x 9 = \$ 0.00
INDEPENDENT	3 - 3 =	0	x78 =\$ 0.00	or	x 39 = \$ 0.00
MULTIPLE DEPENDENT CLAIM PRESENTED <u>no</u>			+260 = \$ 0.00	or	+130 = \$ 0.00
TOTAL \$ 760.00				TOTAL \$ 0.00	

X A check in the amount of \$ 800.00 to cover the filing fee and recording fee (if applicable) is enclosed.

____ Please charge Deposit Account No. 02-2448 in the amount of \$ _____. A triplicate copy of this transmittal form is enclosed.

____ No fee is enclosed.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37 C.F.R. 1.16 or under 37 C.F.R. 1.17; particularly, extension of time fees.

Respectfully submitted,

BIRCH, STEWART, KOLASCH & BIRCH, LLP

By _____

MICHAEL K. MUTTER

Reg. No. 29,680

P. O. Box 747

Falls Church, Virginia 22040-0747

(703) 205-8000
MKM/sas

A SYSTEM OF DYNAMIC MODULE CONFIGURATION AND A METHOD THEREOF

BACKGROUND OF THE INVENTION

Field of the Invention

5 This invention relates to a system wherein a client and a server are linked to a network. Particularly, it is effective to apply the present invention to a system wherein a resource such as a memory is very limited because the server of the system is included in a device of electric products or office facilities. Moreover, the present invention relates to a system for dynamically composing a module at a client-server system wherein the server responds to the client request by making a response using the module in the server.

Description of the Related Art

15 Fig. 12 shows a configuration of a conventional client-server system. A client 1 making a request, a server 2 responding to the client request, a network 3 to which the client 1 and the server 2 are linked, and plural function executing modules 4 each of which makes a response to each request coming from the client 1, are shown in Fig. 12. Contents and services which the client 1 can request are statically specified based on the kinds and the number of the function executing modules stored in the server 2.

25 The system as shown in Fig. 12 has a problem that the kinds and the number of the function executing modules 4 which make responses to requests coming from the client 1, are restricted based on the resource limit, such as a memory amount, of the server 2. Namely, contents and services which the server 2 can provide to the client 1 are statically specified in advance, depending upon the function

executing modules 4 stored in the server 2. Therefore, it is impossible to provide other contents and services except the above.

SUMMARY OF THE INVENTION

5 The present invention is contrived to solve the above-stated problem. It is an object of the present invention to provide a system in which various contents and services can be supplied to the client, by dynamically acquiring a function executing module corresponding to each request of the client and by making the server have only a basic module for acquiring the function executing module, even
10 when the server resource is limited or the server is included in a device whose resource such as a memory is very limited.

According to one aspect of the present invention, a dynamic module configuration system, linked through a network, comprises

a memory, linked to the network, for memorizing a plurality of function
15 executing modules which execute specific processes,

a request device which outputs an execution request for executing one of the specific processes to the network, and

an execution device for receiving the execution request output from the request device through the network, acquiring one of the plurality of function
20 executing modules which has a function of realizing the execution request from the memory through the network, and executing an acquired function execution module.

According to another aspect of the present invention, a dynamic module configuration method, using a network, comprises the steps of

25 memorizing a plurality of function executing modules for executing specific

processes,

outputting an execution request for executing one of the specific processes to the network, and

receiving the execution request through the network, acquiring one of the plurality of function executing modules which has a function of realizing the execution request through the network, and executing an acquired function execution module.

According to another aspect of the present invention, a dynamic module configuration system comprises

an internal resource of a device for performing an original function of the device, and

an execution device for

receiving an access request which requests information in the device,

acquiring one of a plurality of function execution modules, from an

external resource, which has a function of realizing the access request, and

executing an acquired function execution module,

wherein the receiving, acquiring and executing are performed by using a part of the internal resource.

The above and other objects, features, and advantages of the invention will be more apparent from the following description when taken in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings,

Fig. 1 shows a system configuration according to Embodiment 1 of the present invention;

Fig. 2 shows a concrete system configuration of Fig. 1;

Fig. 3 is a flowchart showing processes performed in a server;

5 Fig. 4 is a flowchart showing processes performed in the server;

Fig. 5 is a flowchart showing processes performed in a module storing server;

Fig. 6 shows another system configuration of the present invention;

Fig. 7 shows a concrete system configuration of Embodiment 2 of the present invention;

10 Fig. 8 is a flowchart showing processes performed in the server;

Fig. 9 is a flowchart showing processes performed in the server;

Fig. 10 is a flowchart showing processes performed in the module storing server;

Fig. 11 shows another system configuration of the present invention; and

Fig. 12 shows a conventional client server system configuration.

15 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiment 1.

Fig. 1 shows a system configuration of Embodiment 1. The followings are shown in Fig. 1: a network 8, a plurality of function executing modules $9_0...9_N$ for
20 executing specific processes, a memory 70 linked to the network 8 for memorizing the function executing modules $9_0...9_N$, a request device 50 for outputting a request for executing a specific process to the network 8, an execution device 60 for receiving the execution request output from the request device 50 through the network 8, acquiring one of the function executing modules $9_0...9_N$ from the
25 memory 70 through the network 8 and executing the acquired function execution

module 9, and a device 200 including the execution device 60.

As a concrete example of the device 200, anything such as the followings can be the device 200: printer, facsimile, disk drive, personal computer, telephone, television, radio, air conditioner, refrigerator, automobile, airplane, train,
5 machine tool, elevator, escalator, monitoring camera, etc. Namely, anything which is accessed from the outside to acquire the status or information of the thing can be the device 200. Generally, the device 200 includes internal resources in order to perform the device original function. For instance, the internal resources mean hardware (not shown) of the device and a controller (not shown) of
10 the hardware.

The controller usually has a central processing unit (CPU) and a memory. Each part hardware in the device is controlled by executing a program, which is stored in the memory, at the CPU to perform the device original function. In order to keep the cost of the device 200 low, the least amount of hardware
15 necessary for performing the device original function, the CPU having the least processing capacity necessary for performing the device original function, and the memory having the least amount necessary for performing the device original function are usually installed in the device 200. Especially, in producing household electric appliances or office automation appliances which are mass-
20 produced, it is tried to have the least amount of useless part so as to lower the cost. Namely, the mass-produced device usually has the least amount of internal resource necessary for performing the device original function.

According to the present embodiment, the execution device 60 having a server function can be installed in the device 200 even when the device 200 does not have
25 enough internal resources. The server function of the execution device 60

installed in the device can be thoroughly operated.

According to the system of the present Embodiment, in order to lessen the internal resources necessary for installing the execution device 60 and executing the server function, one of the function executing modules $9_0 \dots 9_N$ is acquired as a function executing module 9 from the memory 70 for executing the request coming from the request device 50. Then, the acquired function executing module 9 is executed in the execution device 60, and the execution result is transmitted to the request device 50. In the present system, the function executing module 9 is not stationed at the execution device 60 but stationed at the memory 70. The memory 70 is an external resource being independent of the device 200. Only a module for acquiring one function executing module 9 from the memory 70 and executing the acquired function executing module 9 is installed in the execution device 60. At every request coming from the request device 50, one function executing module 9 corresponding to the request is acquired and is put into the execution device 60. Because of this configuration, the size of the internal resource used for the execution device 60 in the device 200 can be small. Accordingly, it is possible to install the execution device 60 in the device 200 having limited internal resources, such as a household electric appliance, and to execute a requested function executing module 9 by using the limited internal resources in the device 200.

Fig. 2 shows a concrete configuration system of Fig. 1. The followings are shown in Fig. 2: a client 5 (an example of the request device 50) which requests contents, a server 6 (an example of the execution device 60) for sending the requested contents to the client 5, a module storing server 7 (an example of the memory 70) for storing modules which have functions necessary for the server 6 so

as to send the requested contents to the client 5, a printer 20 (an example of the device 200) including the server 6, the network 8 to which the client 5, the server 6 and the module storing server 7 are linked, the function executing modules $9_0 \dots 9_N$ which are stored in the module storing server 7 and have functions needed when the server 6 sends the contents to the client 5, a contents-request receiving module 10 for receiving the contents-request from the client 5, a contents-request analyzing module 11 for analyzing the contents-request received by the contents-request receiving module 10 in order to select one of the function executing modules $9_0 \dots 9_N$ which has a function needed in sending the content to the client 5, a module requesting module 12 for requesting the selected function executing module 9 from the module storing server 7 and receiving the requested function executing module 9 from the module storing server 7, a module executing module 13 for executing the function executing module 9 received by the module requesting module 12, a module-request receiving module 14 for receiving the module-request from the module requesting module 12, a module acquiring module 15 for acquiring a requested function executing module 9 out of the function executing modules $9_0 \dots 9_N$ based on the module request received by the module-request receiving module 14, and a module transmitting module 16 for transmitting one of the function executing modules $9_0 \dots 9_N$ acquired by the module acquiring module 15 to the server 6.

Detailed explanation for the processes of the client 5, the server 6 and the module storing server 7 will be described with reference to the flowcharts of Figs. 3, 4 and 5, and the arrows between modules in Fig. 2.

Figs. 3 and 4 show the processes performed at the server 6 and Fig. 5 shows those at the module storing server 7.

The server in the present embodiment is different from a server in the general client-server system. The server of the present embodiment does not mean the one having independent hardware and independent software prepared in advance, but means the one later installed in a device having a specific original function.

Namely, the sever of the present embodiment can be realized by borrowing and using the device internal resources provided in order to accomplish the specific original function. For instance, the server 6 is realized as a program stored in the controller memory (not shown) of the printer 20. Then, when the program is executed by the controller CPU, the processes of the server 6 shown in Figs. 3 and 4 can be executed. In the case of the client 5 inquiring the status of the printer 20, for example, being printed or not, or inquiring the print waiting status, the server 6 installed in the printer 20 starts the following processes:

First, the contents-request receiving module 10 in the server 6 is waiting for a contents-request coming from the client 5. (S31) The client 5 requests contents from the sever 6. (arrow ①) The contents-request receiving module 10 receives the contents-request from the client 5 and sends the contents-request to the contents-request analyzing module 11. (arrow ②)

Then, the contents-request analyzing module 11 analyzes the received contents-request in order to select one of the function executing modules $9_0...9_N$ which has a necessary function for sending contents to the client 5, and informs the module requesting module 12 of the selected function executing module 9. (S32) (arrow ③)

The module requesting module 12 requests the selected function executing module 9 from the module storing server 7. (S33) (arrow ④)

In the meantime, the module-request receiving module 14 in the module storing server 7 is waiting for the module request from the server 6. (S51) After receiving the module request from the server 6, the module-request receiving module 14 sends it to the module acquiring module 15. (arrow ⑤)

5 Then, receiving the module request, the module acquiring module 15 acquires the requested function executing module 9 based on the module request (S52) (arrow ⑥) and sends the acquired module to the module transmitting module 16. (arrow ⑦)

10 The module transmitting module 16 transmits the received function executing module 9 to the sever 6. (S53) (arrow ⑧)

In the meantime, the module requesting module 12 has been waiting for the coming of the requested function executing module 9 since the module requesting module 12 requested the function executing module 9 from the module storing server 7. (S34)

15 After receiving the requested function executing module 9 (arrow ⑧), the module requesting module 12 sends it to the module executing module 13. (arrow ⑨) The module executing module 13 actuates a different thread shown in Fig. 4 in order to execute the requested function executing module 9 (S35). Then, the process-flow control goes back to the contents-request receiving module 10 to wait
20 for another contents-request for repeating the above processes again. Instead of actuating the different thread for executing the requested function executing module 9, it is also acceptable to execute the requested function executing module 9 in a new process or in the present process.

In the different thread, the requested function executing module 9 is started
25 (arrow ⑩) and it is waited until the requested function executing module 9 has

been finished. (S41)

Then, the function executing module 9 performs a process of sending the contents to the client 5, which is the completion of the process. (arrow ⑪)

In the different thread, when the function executing module 9 has been finished, the module 9 is deleted from the server 6. (S42)

The above is the way the printing status or print waiting status of the printer 20 is sent to the client 5.

As stated above, the followings are features of the present embodiment system: The client 5, the server 6 and the module storing server 7 are linked to the network 8. The server 6 acquires one of the function executing modules $9_0...9_N$ for performing a process corresponding to a request coming from the client 5, out of the module storing server 7, executes the acquired function executing module 9 and deletes the function executing module 9 from the server 6 when the function executing module 9 has been finished.

According to the above system, it is possible to supply various services by using a memory of small amount, because the server 6 acquires one of the function executing modules $9_0...9_N$ for performing a process corresponding to each contents-request relating to the device 200 from the client 5, out of the module storing server 7.

The function executing modules $9_0...9_N$ are stored in the module storing server 7 in Embodiment 1. It is also acceptable, as shown in Fig. 6, to provide the function executing modules $9_0...9_N$ in the client 5.

In the system of Fig. 6, the client 5 and the server 6 are linked to the network 8. When a request comes from the client 5, the server 6 acquires one of the function executing modules $9_0...9_N$ to perform a process corresponding to the

request, executes the acquired function executing module 9 and deletes the module 9 from the server 6 when the module 9 has been finished.

Embodiment 2.

Fig. 7 shows a concrete system according to Embodiment 2. The elements 5 through 16 and 20 in Fig. 7 are the same as those in Embodiment 1. A module storing module 17 stores acquired function executing modules 9 from the module storing server 7 as many as possible in the resource of the server 6. A module caching module 18 caches the requested function executing module 9 after sending the requested function executing module 9 to the server 6 based on the module request.

Detailed explanation for the processes of the client 5, the server 6 and the module storing server 7 will be described with reference to the flowcharts of Figs. 8, 9 and 10, and the arrows between modules in Fig. 7.

Figs. 8 and 9 show the processes performed at the server 6 and Fig. 10 shows those at the module storing server 7.

First, the contents-request receiving module 10 in the server 6 is waiting for a contents-request from the client 5. (S31) The client 5 requests contents from the sever 6. (arrow ①)

Then, the contents-request receiving module 10 receives the contents-request from the client 5 and sends the contents-request to the contents-request analyzing module 11. (arrow ②)

The contents-request analyzing module 11 analyzes the received contents-request in order to select one of the function executing modules $9_0 \dots 9_N$ which has a necessary function for sending contents to the client 5, and informs the module

requesting module 12 of the selected function executing module 9. (S32) (arrow ③)

The module requesting module 12 checks the module storing module 17 to find the selected function executing module 9 is stored or not. (S81) (arrow ④)

5 If the selected function executing module 9 is stored, the module storing module 17 transmits the selected function executing module 9 to the module requesting module 12. If not stored, the module storing module 17 informs the module requesting module 12 that the selected function executing module 9 is not stored. (arrow ④)

10 In the case of the selected function executing module 9 being transmitted from the module storing module 17, the module requesting module 12 transmits it to the module executing module 13. If it is informed that the selected function executing module 9 is not stored in the module storing module 17, the module requesting module 12 requests the selected function executing module from
15 module storing server 7. (S33)

In the meantime, the module-request receiving module 14 in the module storing server 7 is waiting for the module request from the server 6. (S51)

After receiving the module request from the server 6 (arrow ⑤), the module-request receiving module 14 sends the module request to the module
20 acquiring module 15. (arrow ⑥)

Then, receiving the module request, the module acquiring module 15 checks the module caching module 18 to find whether the requested function executing module 9 is cached or not. (S101) (arrow ⑦)

If the requested function executing module 9 is cached, the module caching
25 module 18 sends it to the module acquiring module 15. If not cached, the module

5 caching module 18 informs the module acquiring module 15 that the requested function executing module 9 is not cached. (arrow ⑦)

10 In the case of the selected function executing module 9 being transmitted from the module caching module 18, the module acquiring module 15 transmits it to the module transmitting module 16. (arrow ⑩) If it is informed that the requested function executing module 9 is not cached, the module acquiring module 15 acquires the requested function executing module 9 (S52) (arrow ⑧), transmits it to the module caching module 18 and module transmitting module 16. (arrows ⑨ and ⑩) Then, the module caching module 18 caches the transmitted function executing module 9. (S102)

The module transmitting module 16 transmits the received function executing module 9 to the sever 6. (S53) (arrow ⑪)

15 In the case that the module requesting module 12 requested the function executing module 9 from the module storing server 7, the coming of the requested function executing module 9 has been waited by the module requesting module 12. (S34)

20 After receiving the requested function executing module 9 (arrow ⑪), the module requesting module 12 sends it to the module executing module 13. (arrow ⑬) In the case of the requested function executing module 9 being sent from the module storing server 7, the module requesting module 12 also sends the module 9 to the module storing module 17. Then, the module storing module 17 stores the sent function executing module 9 if there is space for it in the resource of the server 6. If there is no enough space, the oldest function executing module 9 in the resource is deleted for making space to store the just sent function executing module 9. Consequently, the just sent function executing module 9 is stored.

(S84) (arrow ⑫) FIFO (First In First Out) is used as the storing algorithm in the above, and it is also acceptable to apply other algorithms.

The module executing module 13 actuates a different thread in order to execute the requested function executing module 9. (S35) Then, the process-flow control goes back to the contents-request receiving module 10 to wait for another contents-request for repeating the above processes again. Instead of actuating the different thread for executing the requested function executing module 9, it is also acceptable to execute the requested function executing module 9 in a new process or in the present process.

10 In the different thread, the requested function executing module 9 is started (arrow ⑭) and it is waited until the requested function executing module 9 has been finished. (S41)

Then, the function executing module 9 performs a process of sending the contents to the client 5, which is the completion of the process. (arrow ⑮)

15 In the different thread, when the function executing module 9 has been finished, the module 9 is deleted from the server 6. (S42)

As stated above, the followings are features of the present embodiment system: After the module execution has been finished, the executed module is stored as long as there is space for storing it in the resource of the server 6, for the purpose of using the same module again.

As the module storing server 7 caches modules acquired by the server 6, in the case of using the same module again, one of the cached modules can be picked up from the cache and can be sent to the server 6.

According to the above system, the following advantages are achieved: It is possible to supply various services by using a memory of small amount, because

the server 6 acquires one of the function executing modules $9_0...9_N$ for performing a process corresponding to each contents-request from the client 5, out of the module storing server 7. Moreover, as the function executing modules are stored in the server or cached in the module storing server, the speed of responding to the client is increased and data transmitted to the network or data used for linking the sever and the module storing server is decreased.

The function executing modules $9_0...9_N$ are stored in the module storing server 7 in Embodiment 2. It is also acceptable, as shown in Fig. 11, to provide the function executing modules $9_0...9_N$ in the client 5.

In the system shown in Fig. 11, when the module execution has been finished, the executed module is stored as long as there is space for storing it in the resource of the server 6, for the purpose of using the same module again.

As the client 5 caches modules acquired by the server 6, in the case of using the same module again, one of the cached modules can be picked up from the cache and can be sent to the server 6.

The modules described in the above Embodiments can be hardware, software, firmware or combination of them.

Having thus described several particular embodiments of the invention, various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications, and improvements are intended to be part of this disclosure, and are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only, and not intended to be limiting. The invention is limited only as defined in the following claims and the equivalents thereto.

What is claimed is:

1. A system of dynamic module configuration which is linked through a network comprising:

a memory, linked to the network, for memorizing a plurality of function

5 executing modules which execute specific processes;

a request device which outputs an execution request for executing one of the specific processes to the network; and

an execution device for receiving the execution request output from the request device through the network, acquiring one of the plurality of function
10 executing modules which has a function of realizing the execution request from the memory through the network, and executing an acquired function execution module.

2. The system of dynamic module configuration of claim 1, wherein the execution
15 device deletes the acquired function execution module after the acquired function execution module has been executed.

3. The system of dynamic module configuration of claim 1, wherein the execution
20 device stores the acquired function execution module after the acquired function execution module has been executed, and re-executes the acquired function execution module stored in the execution device when it is requested to execute a module having a function corresponding to the acquired function execution module.

25 4. The system of dynamic module configuration of claim 1, wherein the memory

caches the function execution module acquired by the execution device and provides the function execution module cached in the memory when it is requested to acquire a module, which has a function corresponding to the function execution module cached in the memory, by the execution module.

5

5. The system of dynamic module configuration of claim 1, wherein the request device and the memory are installed in a device.

6. The system of dynamic module configuration of claim 1, wherein the request device is a client which outputs a contents request corresponding to the execution request, the execution device is a server which receives the contents request and responds to the contents request, and the memory is a module storing server which stores the plurality of function executing modules for responding to the contents request.

10

15

7. The system of dynamic module configuration of claim 6, wherein the server includes

 a contents-request receiving module for receiving the contents request from the client,

20

 a contents-request analyzing module for analyzing the contents request received by the contents-request receiving module in order to select one of the plurality of function executing modules which has a function needed in responding to the contents request,

25

 a module requesting module for requesting a selected function executing module from the module storing server based on an analyzing result by the

contents-request analyzing module, and for receiving the selected function
executing module from the module storing server, and

a module executing module for executing the selected function executing
module received by the module requesting module.

5

8. The system of dynamic module configuration of claim 7, wherein the module
storing server includes

a module-request receiving module for receiving a module request from the
module requesting module,

10

a module acquiring module for acquiring a function executing module out of
the plurality of function executing modules based on the module request received
by the module-request receiving module, and

a module transmitting module for transmitting the function executing
module acquired by the module acquiring module to the server.

15

9. The system of dynamic module configuration of claim 7, wherein the server
further includes a module storing module for storing the selected function
executing module acquired from the module storing server as many as possible in
a resource of the server.

20

10. The system of dynamic module configuration of claim 8, wherein the module
storing server further includes a module caching module for caching the selected
function executing module after the selected function executing module has been
sent to the server.

25

11. A dynamic module configuration method using a network comprising the steps of :

memorizing a plurality of function executing modules for executing specific processes;

5 outputting an execution request for executing one of the specific processes to the network; and

receiving the execution request through the network, acquiring one of the plurality of function executing modules which has a function of realizing the execution request through the network, and executing an acquired function
10 execution module.

12. The dynamic module configuration method of claim 11, wherein the step of executing the acquired function execution module includes the step of deleting the acquired function execution module after the acquired function execution module
15 has been executed.

13. The dynamic module configuration method of claim 11, wherein the step of executing the acquired function execution module includes the step of storing the acquired function execution module after the acquired function execution module
20 has been executed, and re-executing the acquired function execution module when it is requested to execute a module having a function corresponding to the acquired function execution module.

14. The dynamic module configuration method of claim 11, wherein the step of
25 memorizing the plurality of function executing modules includes the step of

5
caching the acquired function execution module, and providing the acquired
function execution module cached at the caching step when it is requested to
acquire a module having a function corresponding to the acquired function
execution module.

15. A system of dynamic module configuration comprising:

an internal resource of a device for performing an original function of the
device; and

an execution device for

10 receiving an access request which requests information in the device,
acquiring one of a plurality of function execution modules, from an
external resource, which has a function of realizing the access request, and
executing an acquired function execution module,
wherein the receiving, acquiring and executing are performed by using a
15 part of the internal resource.

16. The system of dynamic module configuration of claim 15, wherein the internal
resource includes a central processing unit and a memory, the execution device
includes a program stored in the memory and executed by the central processing
20 unit, and the external resource includes a memory, being independent of the
device, for memorizing the plurality of function execution modules.

Fig. 1

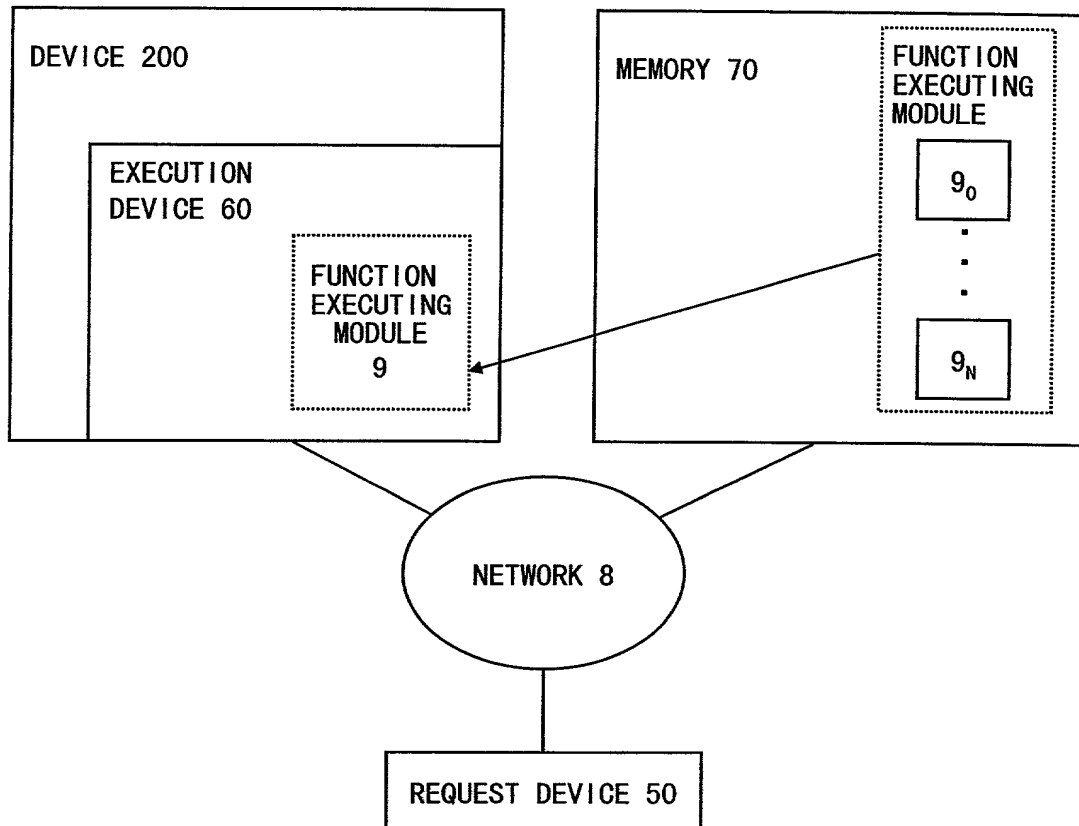


Fig. 3

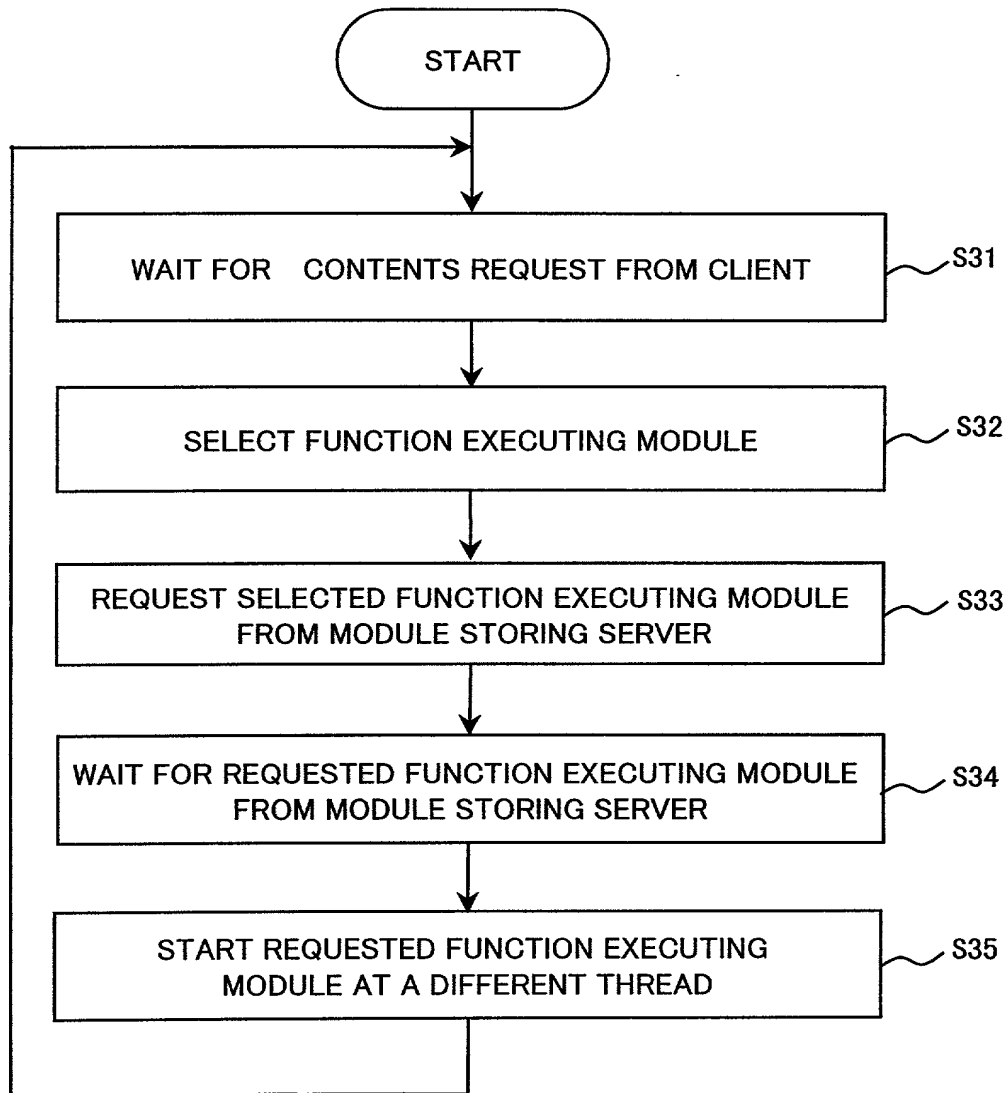


Fig. 4

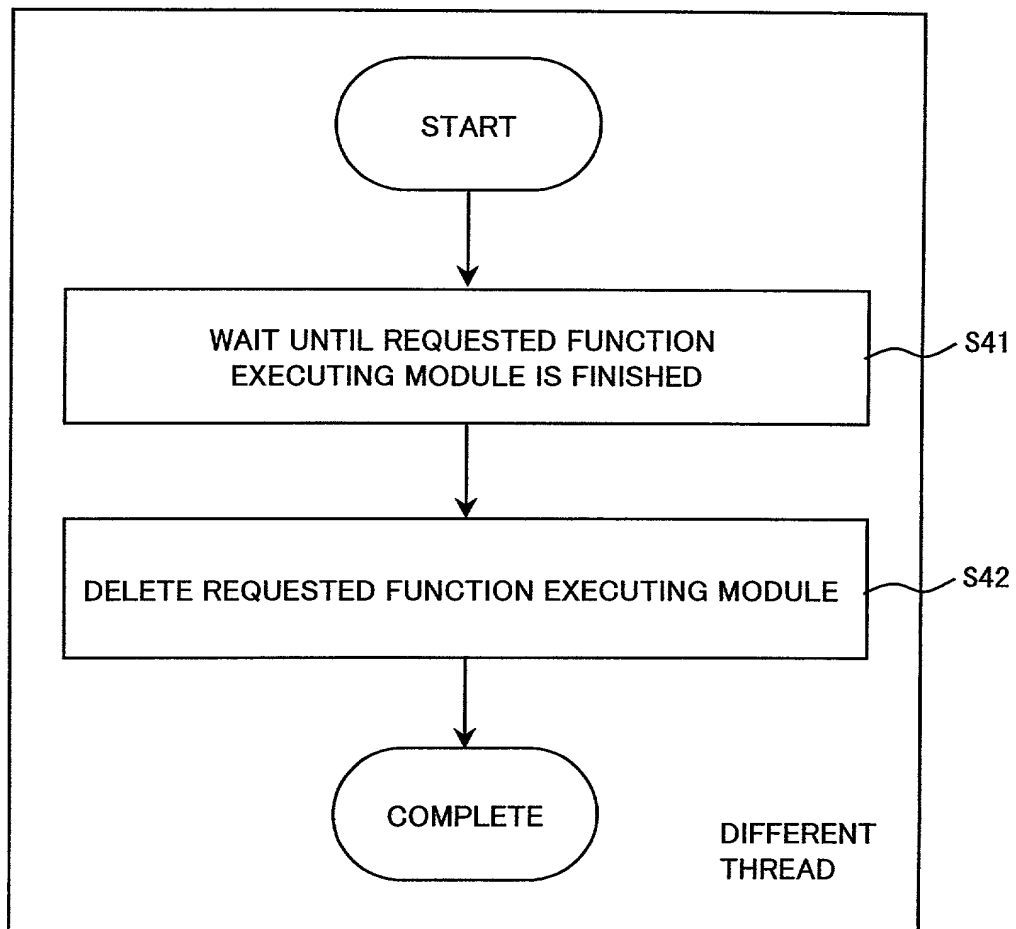


Fig. 5

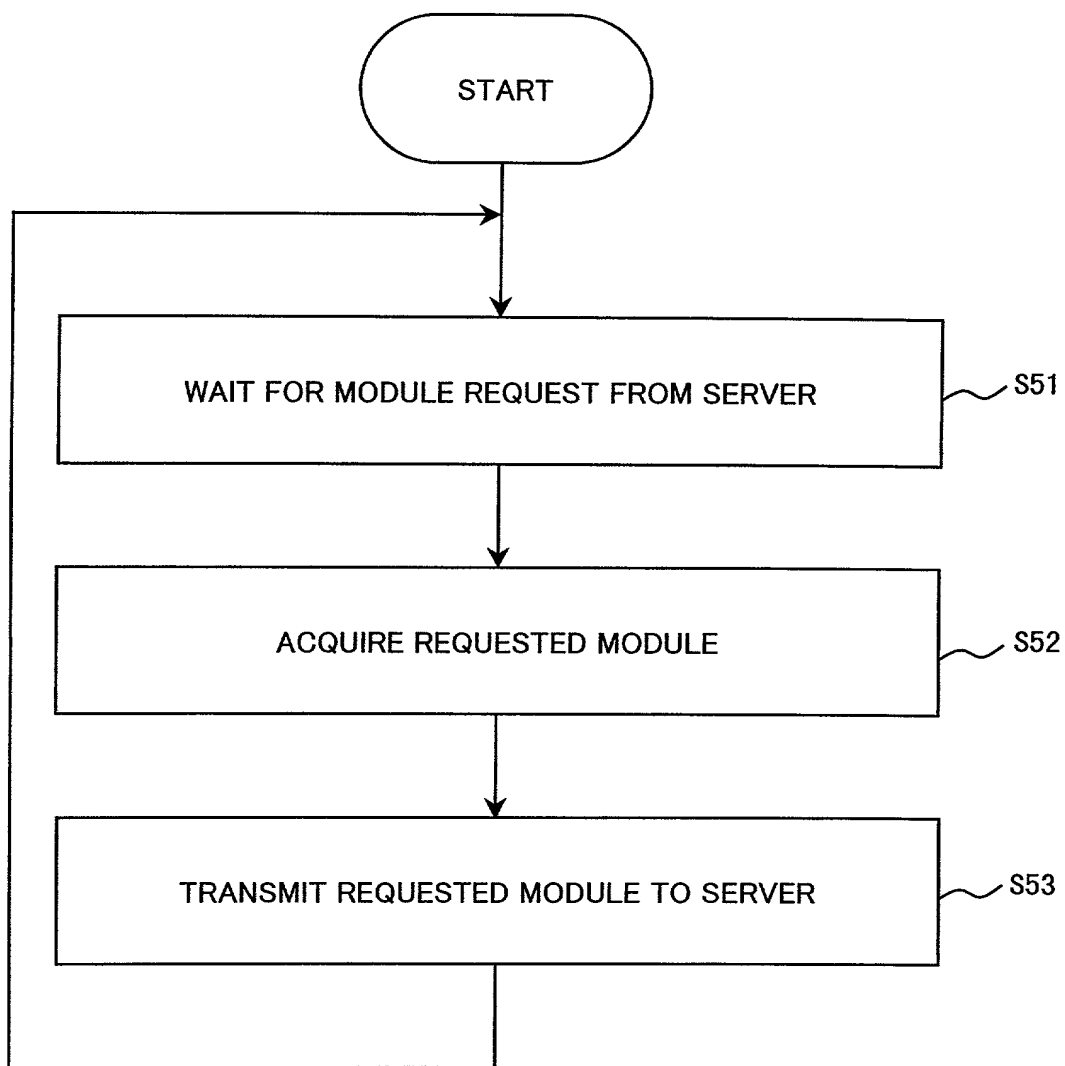


Fig. 6

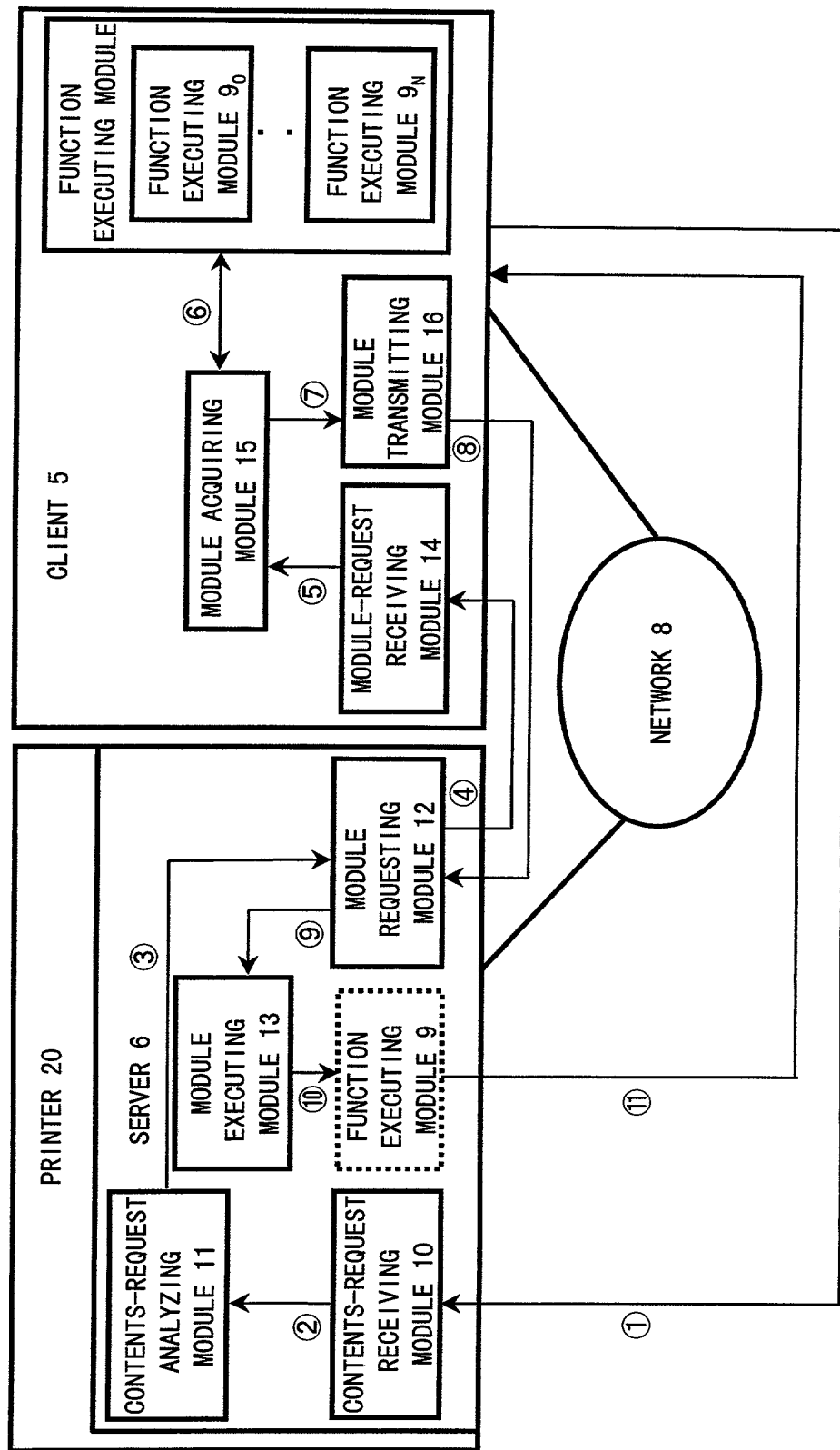


Fig. 7

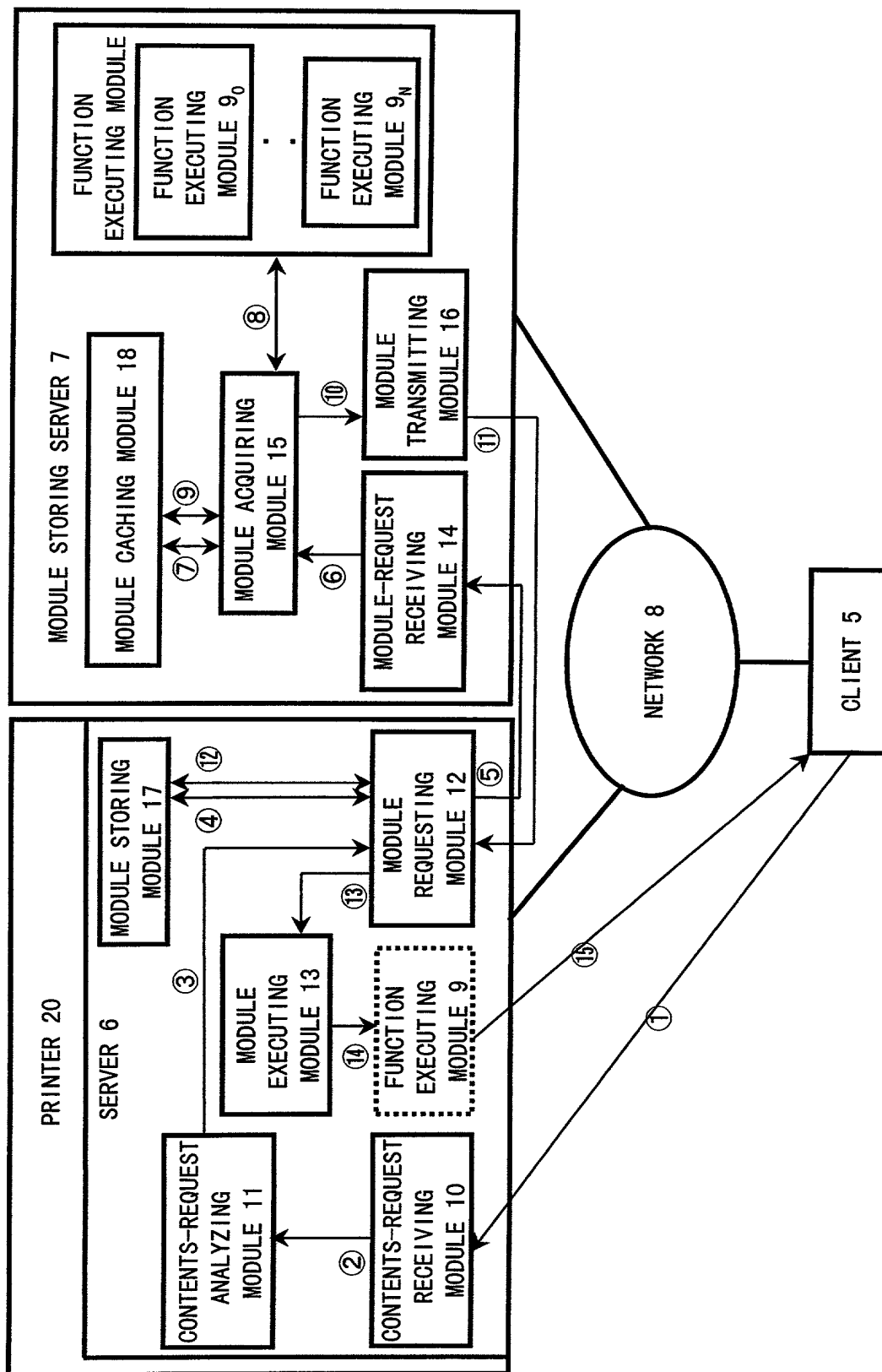


Fig. 8

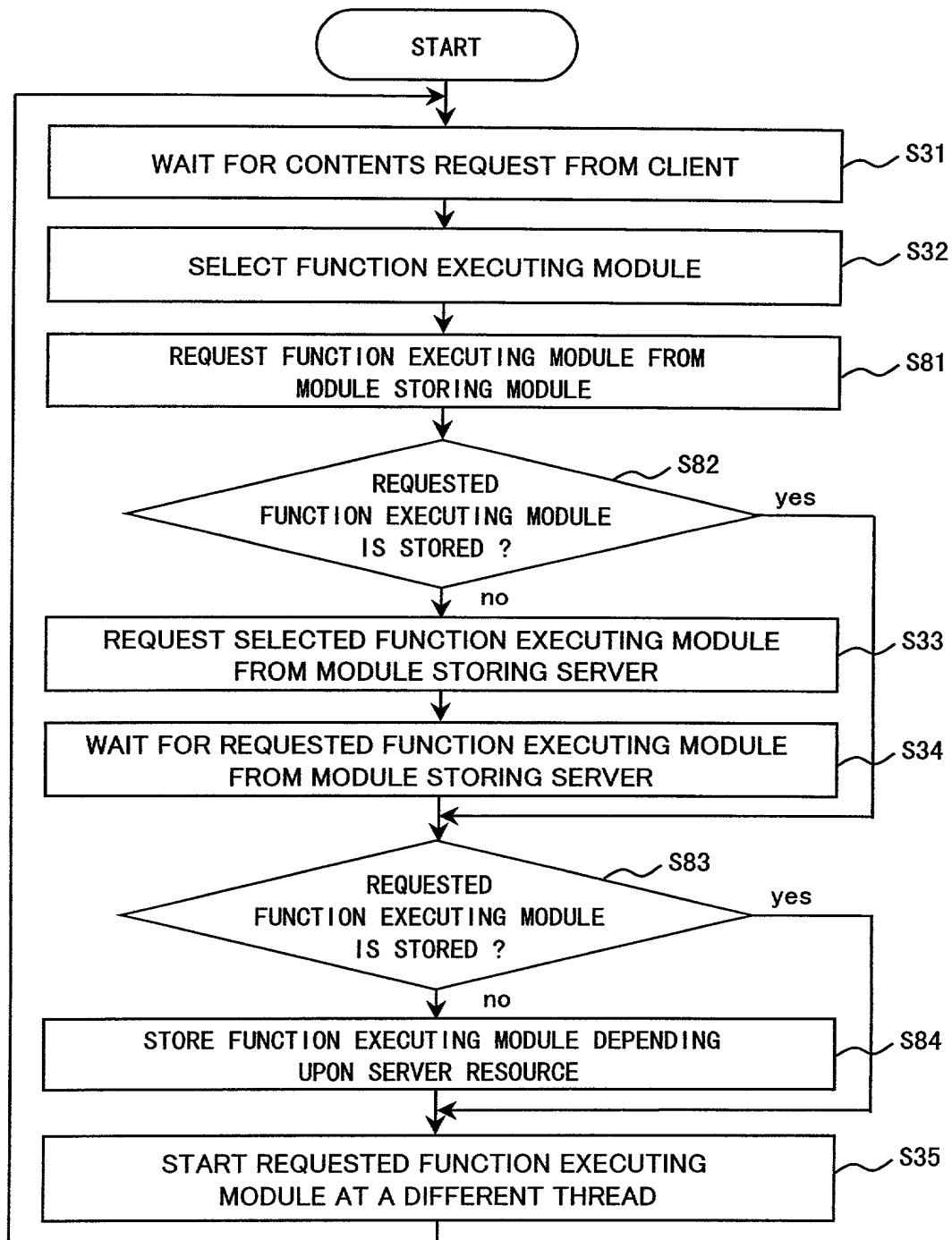


Fig. 9

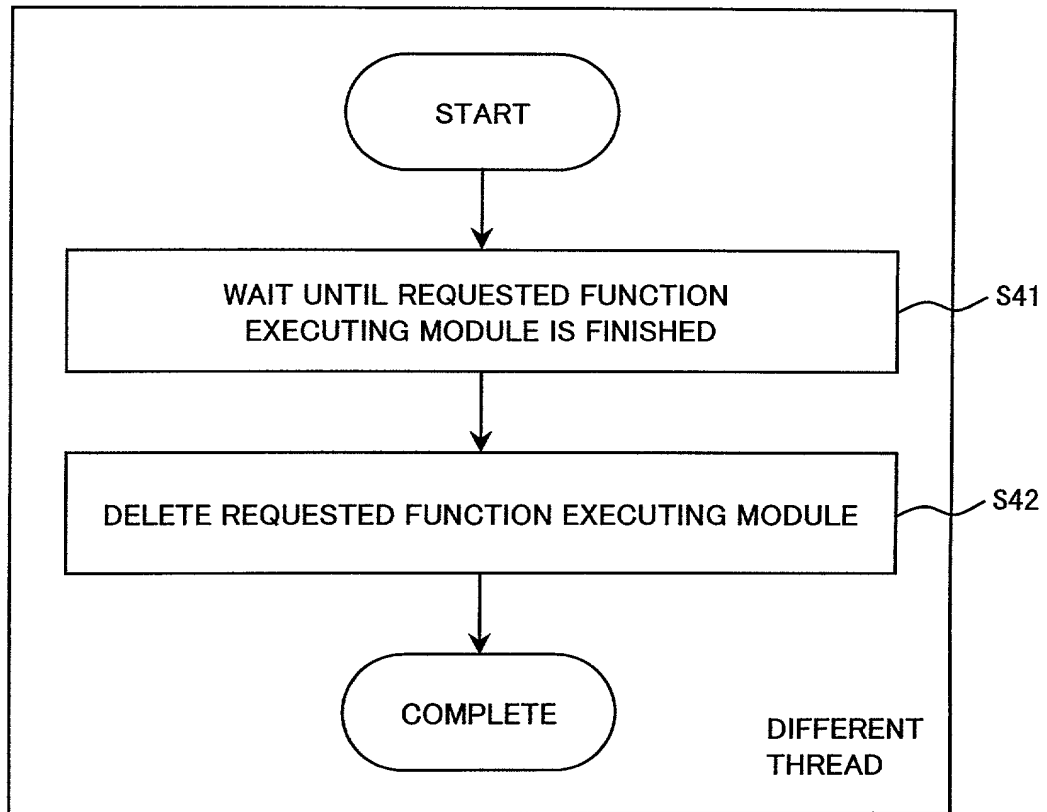


Fig. 10

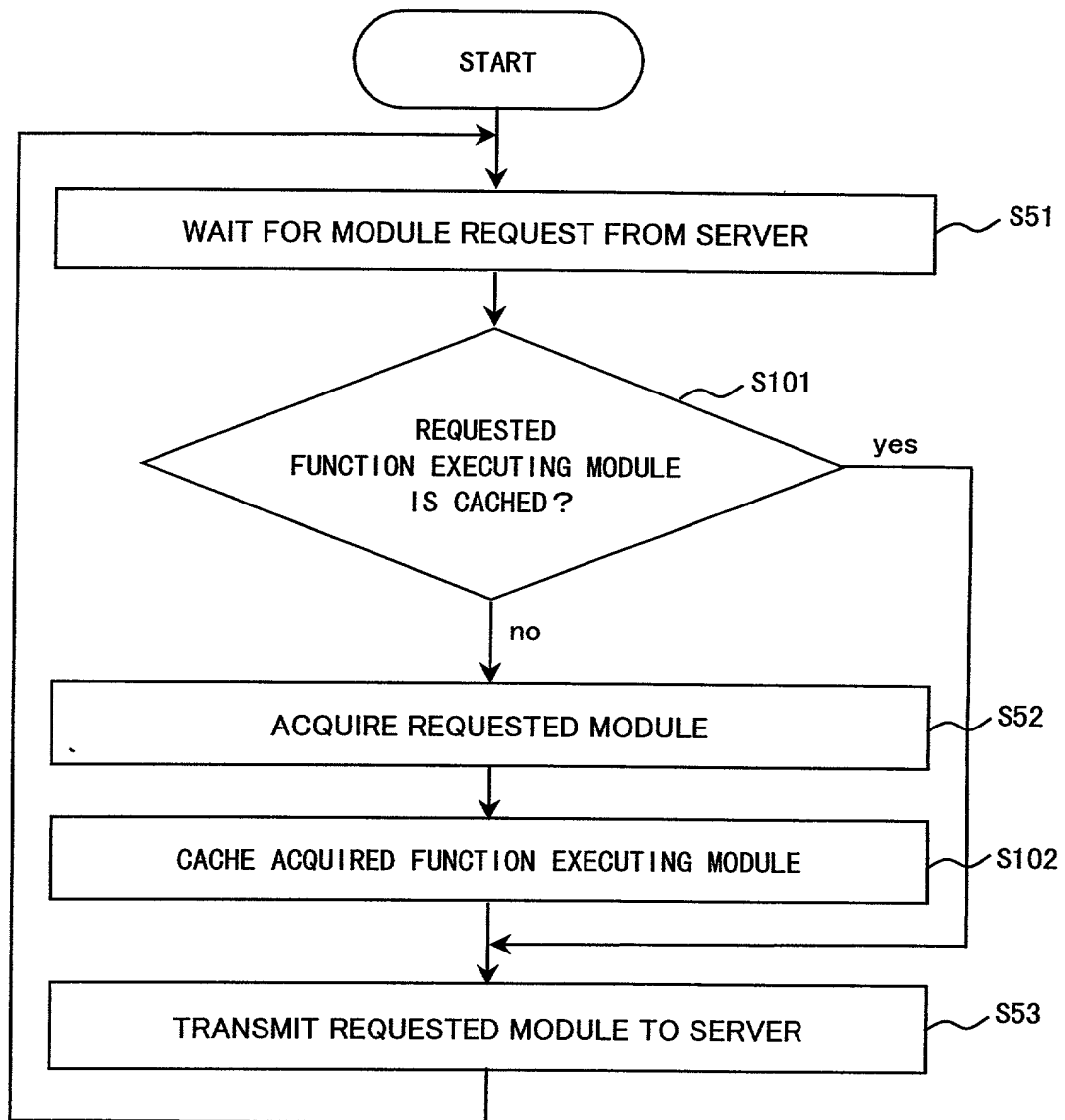
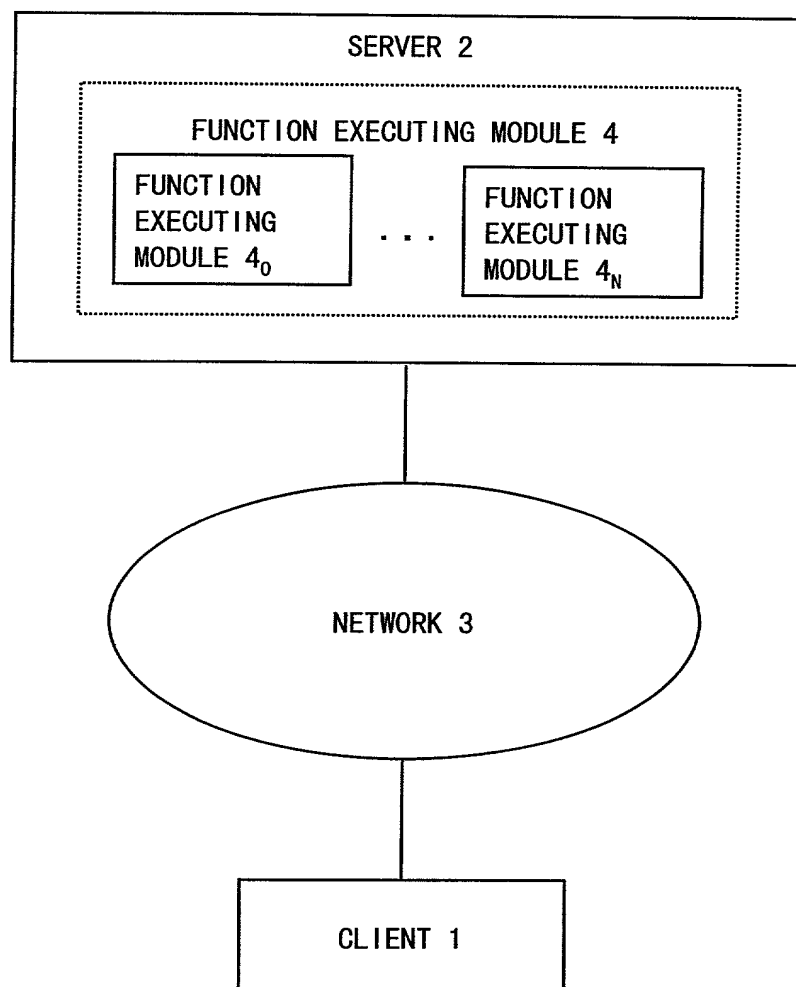


Fig. 12
RELATED ART



Declaration and Power of Attorney for Patent Application**特許出願宣告書及び委任状****Japanese Language Declaration****日本語宣言書**

下記の氏名の発明者として、私は以下の通り宣言します。

As a below named inventor, I hereby declare that:

私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。

My residence, post office address and citizenship are as stated next to my name.

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) or the subject matter which is claimed and for which a patent is sought on the invention entitled

A System of Dynamic Module
Configuration and a Method
Thereof

上記発明の明細書（下記の欄で x 印がついていない場合は本書に添付）は、

The specification of which is attached hereto unless the following box is checked:

☐ 月 日に提出され、米国出願番号または特許協定条約国際出願番号を _____ とし、（該当する場合） _____ に訂正されました。

☐ was filed on _____ as United States Application Number or PCT International Application Number _____ and was amended on _____ (if applicable).

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、連邦規則法典第 37 編第 1 条 5 6 項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

Japanese Language Declaration

(日本語宣言書)

私は、米国法典第35編119条(a)-(d)項又は365条(b)項に基づき下記の、米国以外の国の少なくとも一ヶ国を指定している特許協力条約365(a)項に基づく国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or Inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or Inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

外国での先行出願

Priority Not Claimed

優先権主張なし

HEI 10-318602 Japan
Number (番号) Country (国名)

10 November 1998 ☐
Day/Month/Year Filed (出願の年月日)

Number (番号) Country (国名)

Day/Month/Year Filed (出願の年月日) ☐

私は、第35編米国法典119条(e)項に基づいて下記の米国特許出願規定に記載された権利をここに主張いたします。

I hereby claim the benefit under Title 35, United States Code, Section 119 (e) of any United States provisional application(s) listed below

Application No (出願番号) Filing Date (出願日)

Application No (出願番号) Filing Date (出願日)

私は、下記の米国法典第35編120条に基づいて下記の米国特許出願に記載された権利、又は米国を指定している特許協力条約365条(c)に基づく権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第35編112条第1項又は特許協力条約で規定された方法で先行する米国特許出願に開示されていない限り、その先行米国出願書提出日以降で(c)本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第37編1条56項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT international application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of application.

Application No (出願番号) Filing Date (出願日)

Status Patented, Pending, Abandoned(現況: 特許許可済、係属中、放棄済)

私は、私自身の知識に基づいて本宣言書中で私が行う表明が真実であり、かつ私の入手した情報と私の信じることに基づく表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第18編第1001条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行えば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに上記のごとく宣誓を致します。

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Japanese Language Declaration

(日本語宣言書)

委任状： 私は下記の発明者として、本出願に関する一切の手続きを米特許商標局に対して遂行する弁理士または代理人として、下記の者を指名いたします。(弁理士、または代理人の氏名及び登録番号を明記のこと)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith (*list name and registration number*)

TERRELL C BIRCH (Reg No 19,382)
RAYMOND C STEWART (Reg.No 21,066)
JOSEPH A KOLASCH (Reg No. 22,463)
ANTHONY L. BIRCH (Reg.No 26,122)

JAMES M. SLATTERY (Reg No 28,380)
BERNARD L. SWEENEY (Reg No 24,448)
MICHAEL K MUTTER (Reg No 29,680)
CHARLES GORENSTEIN (Reg.No 29,271)

GERALD M MURPHY (Reg.No. 28,977)
LEONARD R. SVENSSON (Reg.No. 30,330)
TERRY L. CLARK (Reg.No 32,644)
ANDREW D. MEIKLE (Reg No. 32,868)

MARC S. WEINER (Reg No. 32,181)
ANDREW F. REISH (Reg No. 33,443)
JOE M. MUNCY (Reg No. 32,334)
C. JOSEPH FARACI (Reg.No 32,350)

書類送付先：

Send Correspondence to

BIRCH, STEWART, KOLASCH & BIRCH, LLP
P O BOX 747
FALLS CHURCH, VA 22040-0747
TEL (703) 205-8000

直接電話連絡先：(名称及び電話番号)

Direct Telephone Calls to: (*name and telephone number*)

BIRCH, STEWART, KOLASCH & BIRCH, LLP
TEL (703) 205-8000

唯一のまたは第1の発明者名	Full Name of sole or first Inventor
同発明者の署名	Tomohisa Yamaguchi
日付	First inventor's signature Date
住所	Tomohisa Yamaguchi May 26, 1999
国籍	Residence
私書箱	Tokyo, Japan
第2共同発明者の氏名	Citizenship
第2発明者の署名	Japan
日付	Post Office Address
住所	c/o Mitsubishi Denki Kabushiki Kaisha
国籍	2-3, Marunouchi 2-chome, Chiyoda-ku,
私書箱	Tokyo 100-8310 Japan
	Full Name of second joint inventor
	Second inventor's signature Date
	Residence
	Citizenship
	Post Office Address

(第三以降の共同発明者についても同様に記載し、署名をすること)

(Supply similar information and signature for third and subsequent joint inventors.)